Distributed Agile Development: 
A Survey of Challenges and Solutions

Harneet Kaur, Hisham M. Haddad, and Jing (Selena) He 
Department of Computer Science, Kennesaw State University, Kennesaw, GA, USA
fharneet@students.kennesaw.edu, {hhaddad, jhe4}@Kennesaw.edu

Abstract—Global market, constant pressure, and global talent are major powers that lead to distributed teams. But this is not an easy task; companies have to deal with several non-trivial geo-political constraints, such as the willingness of their employees to relocate, the costs of such relocations, procedural constraints such as work permit issues, and others. To circumvent this, companies set up their offices in multiple locations and hire local employees who work with their colleagues at different geographical locations. Agility at Scale 2012 survey found that 57% of the respondents were having geographically dispersed teams. Having your resources in different locations may make implementing certain agile processes harder. However, there are ways of working around it. This paper investigates the challenges faced by distributed agile teams and proposes solutions to address these issues through an exploratory literature review. The proposed solutions will help to build successful agile teams.

Keywords: Distributed agile development, Agile practices, Outsourcing, Distributed agile challenges, Distributed agile solutions.

1. Introduction

In the most recent decade, there is a continuous investment to transform local markets to global market. Many different challenges have to be managed like additional faults in software projects and lack of sufficient assets. Software organizations use Distributed Software Development (DSD) amenities to deal with these problems. These amenities help minimize expenses and the way into skilled labour. Their primary goal is to build up worth goods at reduced prices than the co-located developments by enhancing resources. At times, the quest for extreme benefit takes organizations to seek outside solutions in different nations and this is referred to as Global Software Development (GSD).

Software is created at multiple locations, in multiple cultures and in worldwide dispersed locations. The managers, executives, and engineers have to face many challenges at different stages of the development process. These challenges may be social, cultural, or technical. This influences the manner in which software is planned, executed, and conveyed to the customer.

Agile methods work well in notably vibrant industry and IT environment. Organizations are restless in looking for talent and skills accessible at easier rates. And hence the desire to outsource the development process to these countries. The aim of this paper is to understand the challenges faced by the geographically dispersed agile teams and propose practices that can be used to overcome these challenges.

The paper is organized as follows: Section 2 highlights distributed agile development. Section 3 discusses the challenges faced by geographically dispersed agile teams. Section 4 presents solutions for these challenges. The section 5 concludes the paper.

2. Distributed Agile Development

Agile development came into existence in 2001 [1] and was considered to be the foundation to change the software development practices. This was accomplished by mediating the risk of altering needs and evolving technologies. Agility intends to strip away the complexity associated with traditional development. Thus concentrates on its deciding objective to accelerate the project due dates, lift up the brisk reaction to evolving situations and changes in client necessities etc.

Distributed development is considered to be an unavoidable truth for numerous agile teams. Majority of agile methods presume that the teams to be placed in single room but unfortunately this does not go with the real world situations in which the agile teams are distributed globally throughout the world. The factors that give rise to dispersed teams are the following:

1. Global Market: Business market is expanding at a very fast rate and when some new business steps in, it has to match its standards. For that purpose it needs to advance the knowledge in those markets with the help of amalgamation and setting up or gaining subsidiaries situated in those markets.
2. **Global Talent:** In this competitive world, companies are hiring skilled and experienced employees. There are some factors that lead to distribution of teams such as: Willingness of employees to relocate, Availability of Work Visa, and Cost of relocation.

3. **Reducing Costs:** Nowadays companies are outsourcing to areas with economical development rates to reduce the development cost. It is estimated that 25% cost savings are there if the service providers are located offshore than domestic.

   There are several Agile stories reported in the literature [2, 3, 4, 5]. The “Agility at Scale 2012” survey [6], illustrates how distributed were the teams on the successful projects: 49% of the agile teams had team members spread out through the same building; 29% of the team members were working within the driving distance; 30% of the agile team members were working from home; and 52% indicated that some of their agile teams had team members that were far located.

   These facts illustrate that software companies do not go with the idea of whole agile team working together in one room. So, there is a strong requirement to combine Agile practices with distributed development practices. Combining these approaches introduces some challenges which are discussed in the following section.

   Organizations that choose to be both distributed and Agile have different ways of implementing that through the team structure: *Isolated, Semi-integrated, and Integrated* [7]. With *Isolated* structure, the organization may decide to separate all the functionality at a particular location to form isolated teams. The benefit of this approach is that numerous issues related to distributed teams are averted. With *Semi-integrated* structure, every location develops a team and only the set of overlapping features with other teams are dealt. This structure promotes further sharing of knowledge and lesser knowledge silos. Only drawback of this structure is dealing with communication problems. With *Integrated* structure, the team is composed of members from diverse locations. Although this maximizes knowledge sharing, it increases the possibility of miscommunication due to lack of team cohesiveness.

3. **The Challenges**

   Although distributed teams are considered to be more effective than co-located teams because of the reduced cost, global talent, and others, there are certain loopholes in this approach. The following are the challenges faced by distributed agile teams:

### 3.1 Documentation

   Agile teams do not give importance to the documentation. This may affect the distributed teams as they will miss some details about the project and hence their understanding about the project will suffer.

### 3.2 Pair Programming

   Agile development uses pair programming in which two members of the team work on the same code side by side. This approach is totally impossible in distributed environment. Hence distributed groups will have to find some other similar methodology.

### 3.3 Different Work Hours

   Sometimes, there come situations when team members are located in different time zones and their working hours don’t match. Hence their working hours need to be aligned so that they can communicate with each other. This helps avoiding rework and provides clarity of project.

### 3.4 Communication

   Reduced communication has more effects in case of distributed teams. Most agile practices like test driven development can be educated by providing one-on-one training. Many problems in distributed agile development are related to communication like unable to understand the customer, the system architecture or system design. These have to be solved by participating in discussions or solving the problem manually.

### 3.5 Knowledge Transition

   Knowledge transition is absent in project development, processes of customer support, domain and central product. The developing teams have to set up the overlap times for different time zones so as to achieve the 24 hours and 7 days yield.

### 3.6 Cultural Differences

   Cultural issues can cause misunderstanding between team members. Several recent studies [8, 9, 10, 11, 12] have explored the cultural differences and measures to manage them in distributed teams.

### 3.7 Lack of Team Cohesion

   In case of distributed development, members at distinctive locales are more averse to observe themselves as a major aspect of the same group when contrasted with co-placed members. Absence of togetherness, accompanied by common view of goals, is an issue in that situation. They get worse when we talk about agile development because it focuses on regular collaboration on all phases of the software project.
3.8 People vs. Process Oriented Approaches

In agile development, the process is people oriented and informal methods are used to establish the control whereas distributed development needs the control to be achieved by formal methods.

3.9 Knowledge Management

During the development process the experience of team members, decisions, methods and skills must be gathered through knowledge sharing. This helps the team members to use the experience of the precursor to reduce redundant work and cost. The benefit of global distribution cannot be acquired without effective knowledge or information sharing. Hence knowledge should be managed properly in distributed agile development.

3.10 Language Barriers

The language problem arises when the teams are non-collocated and hence they are not able to understand each other due to their different languages.

3.11 Role of Specialist

Typical software organizations always have people with specialized knowledge like business analyst, testers and user interface specialists etc. Their knowledge is expected to be utilized in the project when needed. Agile methodology does not have any formal mechanism to request such expertise. Even if the specialists are brought, they may face problems similar to a new team member about gaining the understanding of the project requirements [13].

3.12 Developer Fear of Skill Deficiency Exposure

Some developers fear that agile processes can bring forward their deficiencies. Onsite customers, stand-up meetings, use of storyboards and whiteboards bring the shortcoming of developers in front of the whole team because agile methodology involves constant communication and collaboration. In addition, continuous integration and automated testing mean that developers can’t hide poor, low-quality code. Exposing the weaknesses of developers can prove counterproductive [14].

3.13 Recruitment Challenges

It is difficult for agile companies to find right people due to lack of agile-specific recruitment policies. There are only few universities or colleges that incorporate agile methods and skills to their programs. Moreover degree programs tend to rely upon either technical or business skills but rarely involve both [14].

4 The Solutions

In the preceding section we discussed many challenges faced by distributed agile development teams. Making a successful appropriated geographically dispersed agile team is to a great extent about balancing the hindrances to communication due to distribution of teams. Actually numerous geographically distributed groups flounder since they attempt to act as if their group is co-spotted and don’t successfully address the extra communication troubles put on them. A large number of the communication problems confronted require commitments from the group to enhance and the support of extra practices and instruments.

Below we propose practises and techniques to help organizations overcome the challenges discussed in the previous section.

4.1 Documentation

Good documentation may also lead to collaboration of agile teams. For example, if the use case diagrams with user stories reduces misunderstandings and hence enhances collaboration in teams. Several tools are used for documentation like issue tracker (Jira) and project management tool (Scrum Works) [15].

4.2 Pair Programming

Pair programming can be achieved by using communication tools, show-and-tell hour (every team member demonstrates his or her work to the entire team and receives feedback) and a daily developer scrum (developers meet briefly to collaborate on technical issues or approaches) etc.

4.3 Different Working Hours

The agile team members working at different locations faces some communication challenges due to different time zones. Although regular scrums and overlapping working hours helps to minimize this problem but still delays are encountered in the work. Because sometimes clarification is required or rework has to be done. Sometimes the changes made by one person affects the work of other persons at different location and these changes are not propagated correctly. The following helps:

- **Developer to developer handshakes** means that the development team should communicate all the changes they have made during their working hours that the team members at other locations should be aware of.
End of day status notes means that every team member should share with others what he/she did during the work hours, build status, and any kind of issues that has to be handled or ignored.

4.4 Communication

Instead of setting up meetings at 15 different locations, it is good to set up impromptu meetings using video conferencing. This saves time, expenses and also provides flexibility to attend meetings at any time. The different categories of tools used to achieve effective communication [16] are discussed as follows:

4.4.1 Social networking tools

Nowadays there are number of social networking tools as well as social softwares available online which allow interactions in groups. Some of them are: Live meetings, email and Video Conferencing etc.

4.4.2 Communication tools

Instant Messaging (IM) is used to get a quick attention of a team member for a short query. There are some IM applications in which the conversation can be stored as a permanent record. This is additionally an extraordinary method of knowing whether a fellow team member is in the workplace, in a gathering and not be irritated, or accessible for discussion.

4.4.3 SCM tools

SCM tools are used to track as well as control the modifications in the software. Some of the SCM tools are: Version controlling tools and Repository.

4.4.4 Bug and issue tracking databases

These are the database that records information related to bugs and issues.

4.4.5 Knowledge centres

Knowledge centres include frequently asked questions as well as technical references.

4.4.6 Collaborative development environments

These environments provide tools for development in teams, for example, worksites and project workspaces. Some of these tools are as follows:

- Visual Studio Team System. It allows team members to perceive the current state of the project as well as update the tasks of the individual.
- Scrum for Team System It puts into practice burndown charts as well as Scrum task boards to aid with tracking and iteration planning. These tools give the distributed teams an experience of a team room.
- SharePoint. It is used for sharing data and recording the team decisions. They also include cameras to capture pictures of whiteboards.
- Dry Erase board technology. DEBT is not only used to write and erase but also to store what you wrote. DEBT also supports add-ons. Team members can also make a digital copy of board’s data which can be used by the later reviewers to track how the final product is achieved.

4.5 Knowledge Transition

It is achieved by using following methodologies [17]:

- Maintain product/process repository: Creating a database to help the development teams in tracking the status of the project, reporting the issues and assigning priorities.
- Focus on well-understood functionality rather than critical new functionality. Creating an atmosphere in which both the developer and the client get used to the process, application and tools
- Short cycle but not time-boxed development. Using short cycle approach, in which 2-3 advancement cycles were permitted to take 2-4 weeks each one, contingent upon the practicality and the setup time required to comprehend the business space.

4.6 Cultural Differences

This can be accomplished by sharing work practices, understanding cultural differences, managing language barriers, rotating team ambassadors, and engendering cultural awareness [18].

4.7 Team Cohesion

This problem can be addressed by building trust. The trust among the team members is very important because of nominal official control. Some practices were utilized to fabricate the trust between the groups, such as the following.

- Frequent visits by distributed partners. The regular meetings between the project manager and the customer were organized in three companies.
• **Sponsor Visits.** During the starting phases senior manager visited the development team to finalize contracts and to establish ground rules. These visits established great amount of trust in teams.

• **Build cohesive team culture:** Creating a firm group society by needing that every group was made up of parts that had advanced former working associations with one another and aggregately controlled all the needed ability.

### 4.8 People vs. Process Oriented

This challenge can be addressed by:

#### 4.8.1 Continuously adjust the process:

Change the practices to Planning iterations to finalize requirements and develop design and Documenting requirements at different levels of formality.

#### 4.8.2 Verify the trust:

Some of the practices to verify the process and quality of the product are summarized as follow [17]:

- **Distributed QA:** The onshore QA team check the offshore development team for acceptable quality.
- **Supplement informal communication with documentation:** Informal communication should be used accompanied by the documentation of critical artefacts.

### 4.9 Knowledge Management

Effective knowledge management is achieved by the following four processes in distributed agile development [19]:

- **Knowledge Generation:** It involves formal training, self-learning, customer collaboration, inception and communities of practice.
- **Knowledge codification:** It involves technical representation, wiki and documentation.
- **Knowledge Transfer:** It involves tools, pair programming, on site customer, discussions etc.
- **Knowledge Application:** It sprints, similar context or problem solving.

### 4.10 Language Barriers

In geographically distributed agile teams, the frequent communication among team members and between client and developer is very important. If they are from different areas with different language then language barriers may arise in the communication. The ways to overcome this problem are discussed below:

- **ESL (English as a Second Language) Course:** This helps in reading, listening, speaking and understanding.
- **Don’t assume understanding:** Check and notice if the colleague does not ask any question than it means he/she did not understand it.
- **Praise colleagues for asking questions:** Employees should be encouraged to ask questions so that they understand properly. They should be praised for being honest about misunderstanding and never allowed to feel inadequate and powerless.
- **Speak slowly and clearly:** The native employees should speak slowly and clearly so that non native can easily understand and have time to ask questions.

### 4.11 Role of Specialist

Specialist knowledge is required irrespective of what software development methodology is being used. For example, an architect may join the project to create the reference implementation and set the technical direction for the project. The agile team members need to have a certain degree of technical understanding and maturity to take on from the architect once the base framework is in place. Some amount of formalism in form of documentation needs to be introduced to record the recommendations and decisions of the specialist [13].

### 4.12 Developer Fear of Skill Deficiency Exposure

The developers should be provided an environment where they feel safe to expose their weaknesses. This can be achieved by:

- Allowing feedback outside the stand-ups to document any fears, issues, or concerns inappropriate for discussion in open forum.
- Making stand-up meetings voluntary for junior developers.
- Assigning mentors to new staff.
- Pair weaker developers with more experienced developers, giving them joint responsibility for requirements [14].

### 4.13 Recruitment Challenges

These challenges can be solved by developing recruiting practices for agile methods to hire people and by putting
newly recruited graduate in agile projects to get hands-on-experience [14].

5 Conclusion

This survey is conducted to uncover the challenges faced by geographically dispersed agile teams and the ways to conquer them. The findings in this work will help organizations to adopt the distributed agile development without worrying about its challenges. The organizations can use the proposed techniques to run successful distributed agile projects. B. Ramesh in [17] discussed whether we can implement distributed agile development and also few challenges and their solutions. J. Sutherland in [7], used the scrum approach to distributed teams but still this approach did not address many of the challenges like building trust, documentation, team distribution and task distribution. D. Batra in [20] discussed the grounded theory to accomplish the challenges related to communication and cultural differences. G. Rodriguez [21] used virtual meetings as the source of communication between the team members. There are many other sources of communication as well that can be applied to distributed teams. The purpose of this literature survey is to bring up all the challenges, the proposed solutions as well as their context at one place. So that the organizations can find the whole picture of different challenges in this paper and can apply the suitable approach in distributed teams.

More in depth research can be done on the techniques used to conquer the challenges in the future and hence distributed teams can have multiple ways out for their problems; allowing distributed teams to successfully build a project with minimal obstacles. Table 1 provides the summary of challenges, proposed solutions and the corresponding context.

Table 1: Challenges, proposed solutions and Context

<table>
<thead>
<tr>
<th>No.</th>
<th>Challenges</th>
<th>Proposed Solutions</th>
<th>Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Documentation</td>
<td>Use Document Management Tools: issue tracker (Jira) and project management tool (Scrum Works).</td>
<td>4 teams practised it: Two of the teams had participants from TelAviv, France and Florida; and the other two teams had participants in TelAviv only</td>
</tr>
<tr>
<td>B</td>
<td>Pair Programming</td>
<td>Use of video conferencing tools or replace this with equivalent practices like Show-and-Tell hour or a Daily Developer Scrum.</td>
<td>General case applied to projects that utilize pair programming</td>
</tr>
<tr>
<td>C</td>
<td>Different Working Hours</td>
<td>Use of Developer to developer handshakes and end of the day status notes.</td>
<td>Studied by majority of the outsourcees (9 companies) come from North America and the majority of the-outsourcers (8 companies) come from Asia (8 companies)</td>
</tr>
<tr>
<td>D</td>
<td>Communication</td>
<td>Use of Tools: Live Meetings, E-mail, Video Conferencing, Instant Messaging, Visual Studio Team System, Scrum for Team System, Share Point, Dry Erase Board, etc.</td>
<td>Studied by 3 organizations practising agile</td>
</tr>
<tr>
<td>E</td>
<td>Knowledge Transition</td>
<td>Set up overlap time for different time zones to get 24 X 7 yield. Apply knowledge transfer mechanism.</td>
<td>Studied by 3 organizations practising agile</td>
</tr>
<tr>
<td>F</td>
<td>Cultural Differences</td>
<td>Engendering cultural awareness, understanding cultural differences, rotating team ambassadors, sharing work practices, and managing language barriers.</td>
<td>18 Agile practitioners from 10 different software organisations in the USA and India</td>
</tr>
<tr>
<td>G</td>
<td>Team Cohesion</td>
<td>Maintain team involvement and cohesion.</td>
<td>Studied by 3 organizations practising agile</td>
</tr>
<tr>
<td>H</td>
<td>People vs. Process Oriented</td>
<td>Addressed by two groups of practices: Continuously adjust the process and Verify the trust.</td>
<td>Studied by 3 organizations practising agile</td>
</tr>
<tr>
<td>I</td>
<td>Knowledge Management</td>
<td>Use of Knowledge Management Techniques: Knowledge Generation, Knowledge Codification, Knowledge Transfer, Knowledge Application.</td>
<td>45 Agile practitioners from 28 different software companies in the USA, India and Australia.</td>
</tr>
<tr>
<td>J</td>
<td>Language Barrier</td>
<td>Speak slowly and clearly, don’t assume understanding, praise others for asking questions, sign up for English as a foreign language course etc.</td>
<td>18 Agile practitioners from 10 different software organisations in the USA and India</td>
</tr>
<tr>
<td>K</td>
<td>Role of Specialist</td>
<td>Need for a specialist, formal documentation</td>
<td>IT solutions organization based out of India serving customers in US</td>
</tr>
<tr>
<td>L</td>
<td>Developer’s Fear</td>
<td>Allow Feedback, making stand-ups voluntary, assigning mentors, pairing</td>
<td>Initially focussed on group discussions in 2008 and then, conducted 17 case studies in 2009, using in-depth interviews with senior personnel</td>
</tr>
<tr>
<td>M</td>
<td>Recruitment Challenges</td>
<td>Developing recruiting practices, assigning agile projects for experience</td>
<td>Initially focussed on group discussions in 2008 and then, conducted 17 case studies in 2009, using in-depth interviews with senior personnel</td>
</tr>
</tbody>
</table>
Acknowledgment

This work is funded in part by the Kennesaw State University’s Office of the Vice President of Research (OVPR) Pilot/Seed Grant, and by the College of Science and Mathematics Interdisciplinary Research Opportunities (IDROP) Program.

References